General Description of Resource Operation

Most of this third and final year of the original ACME grant period has been spent implementing changes and adjustments required to make the ACME system a useful and productive unit for the Stanford Medical School. Currently ACME service hours, whenever feasible, are 7:00 a.m. to 5:00 p.m., and 6:30 p.m. until midnight. Hardware reliability since the last report has improved significantly. We are, however, still experiencing an average of two hardware failures a week and are further implementing recovery procedures for all hardware errors where even a minimal chance exists of achieving proper recovery. Changes have been made by IBM in both the 270X and 270Y data-acquisition hardware and in the 1800 subsidiary computer's interface hardware to the 360. The changes, together with a better understanding of the data-acquisition devices, have resulted in a reliability comparable to that of the files and the central processing unit. The current bottleneck in the overall system is the speed of the large-core memories (8 miscrosecond cycle time). This could be improved, without any software changes by providing more high-speed memory. The shortage of fast-core memory has been aggravated by changing options within the IBM operating system. To obtain greater reliability, we have shifted to the MVT option of OS/360. Studies and discussions with IBM have led us to believe we will have fewer errors in using MVT. The last three months of operation of the MVT-based system have proven this judgement to be correct; we have had few failures of OS interrupt processing and partitioning. These are the areas of IBM's operating system upon which we rely most heavily. Our experience with the processors, such as the Fortran compiler under the operating system, are not that favorable.

Hardware changes: We added a second 2314 disk unit in December; this second unit is now already 50 per cent filled with user data. This means that our user file capacity has grown from approximately 127 million bytes to 263.6 million bytes with a corresponding--and sometimes too rapid--increase in usage.

We also added slowspeed paper tape equipment on the 1800 in order to service instrumentation needs when data rate and duty cycle make an experiment impractical for on-line data acquisition.

The 1800 can now timeshare 16 user lines; this is generally sufficient for four to five users' data moving simultaneously over the lines. We are experimenting to determine how much the load-factor can be increased without affecting our capability to provide reliable service.

The 270Y and 2701 service has been successfully provided to remote sites with a 20,000 sample-per-second transmission speed. The data path, that is the size of a sample, on the 1800 and the 2701 is 16 bits wide; the data path on the 270Y is 8 bits wide.

More need for 1800 facilities seems to exist and can be handled within the current software design by increasing the amount of core memory storage available in the 1800. Currently a limitation of about 6000 samples per second aggregate net-data rate exists in the communication interface from the 1800 to the 360.

Software developments:

Implementation of "external procedures". This permits programs and subroutines to be shared by users in a modular way and also removes the limit on program size in the ACME system other than that of total hardware core and file size. Of course practical limitations due to the speed of the processor remain.

The direct input/output facility of PL/I has been implemented and enables users to write significant information retrieval and patient-record programs without excessive search time. Organization of these files is a problem for which professional assistance is frequently required. Regina Frey, of the ACME staff, is available to assist with the design and layout of problems of this nature.

PROTECT FILE statements have been implemented to provide data-set protection to users who wish it.

A non-PL form of file has been implemented--the text file--which provides for convenient editing of reports, memorandum, and programs themselves under full program control.

A number of major internal changes were made to the file system. The first change eliminated the necessity for copying the index of a file into a temporary location on a disk when the file is opened. This saved space on the disks, removed the limitation on the number of files which could be opened simultaneously, and greatly reduced the time required to open and close files. A change to the hardware format of the file blocks allows checking the ownership of a block in the channel program without reading the block into core. As a result the channel time required to write a block was reduced by one-half (see ACME Note FIO appended to this section).

The computational facilities of the system have been increased by providing double precision arithmetic. The seven-digit precision of the IBM 360 is generally adequate, but a few users were seriously hampered by the lack of more precision in the ACME system. Double precision provides about 16-digit precision for these users.

Improvements to the PL/ACME language include the INITIAL attribute and the capability to write mutiple statements on one line. Both these facilities can decrease program writing time and program compile time.

FREE and ALLOCATE statements pertaining to arrays were implemented to give the user program control over the amount of memory he is using. This provides him a tool to minimize charges based on the amount of memory used, yet permits him to take advantage of the very large potential memory capability inherent in the ACME system.

A number of new string functions round out the facilities provided in PL/I and some work has been done to speed up the processing of string operators, although much remains to be done in this area.

A number of new programs now make it possible to do much file maintenance while the system is on-line without interfering with users. This has enhanced our capability to respond to errors due to either software or hardware problems.

We have continued our emphasis on guaranteeing all files stored on disk. File integrity has been maintained through continued use of software redundancy checks and by a number of analyzer and restoration utility programs. Backup files are created by nightly dumps of the disks to tape.

Our backing up of file storage has proven to be most useful. No loss of data occurred during the last year. Four potentially disastrous failures did happen. Twice loss was prevented by the interlocks in the file system. The other two times the damage was due to operator error, but full recovery was possible from the backup files.

Maintenance and development work on the library programs has had to be retrenched, and is now going on at a somewhat lower level than we feel would be beneficial to the Stanford Medical School. A list of the library routines is appended to this report. Most of these are statistical in nature; a few provide program editing and translation features.

Our engineering group has installed and maintains 22 laboratory interfaces. The group is occasionally engaged in the design of specialized instrumentation to service specific medical requirements. A new terminal switchboard has been installed in the computer room to cope with the larger frequency of user calls and active terminals.

We have been engaged in circuit and software development to make storage displays manufactured by Tektronix and Hewlett-Packard easily interfaceable for ACME users.

A small effort is underway to evaluate the cost and benefits of making a continuous system modelling program (CSMP) available on ACME. It appears that model builders at Stanford would be greatly aided by an interactive facility. Compatibility with batch versions will be important since ACME is not suitable for large simulations.

An assembler for the 1800 has been produced with PL/ACME. We expect to cancel the card equipment presently on the 1800 and load the machine through the 360. This assembly concept is being studied to determine the methods and benefits of providing assembler services for small computers that are connected to ACME. This could increase their research duty cycle and make smaller configuration feasible. A more unified assembly language could also be an aid in obtaining greater independence from a particular manufacturer by increasing software transparency.

A fair amount of effort has gone into the accounting programs in order to prepare for the new recharge policy.

Operating Policies and Procedures

System reliability has become the primary goal during the last year. This is especially important since some of our users collect instrumentation data on a real-time basis and have no protective backup device. Detailed records are kept of the systems operation. Every failure-software, hardware, or power--is logged, analyzed, and discussed in the weekly staff meeting in an effort to devise means of preventing recurrence. Five backup systems are kept at all times in order to prevent errors being introduced during development of software which might affect daily operations.

Flexibility is the key to success to operations in a research computer environment. This is especially true at ACME. The system is constantly upgraded (both software and hardware--including computer input/output devices and in-house designed and built interface equipment).

We have established an enforced system backup to protect user data and system disk packs. All user data packs are dumped to tape three times a week. Two sets of daily tapes are rotated each week; seven weekly sets are rotated every month. The first set of weekly tapes each month is saved permanently, and replaced with a new set of tapes.

All operating procedures are documented by ACME Notes which are distributed to the ACME staff.

The graph showing hardware and software downtime is presented on the next page. This graph shows utilization. Although considerable detail of this utilization is given in Section I-B of this report, it is appropriate at this point to note some of the highlights of utilization of the ACME system. There has been an uptrend ever since the basic software was stabilized enough to present meaningful statistics. After the sharp dips in utilization as the result of hardware difficulties toward the end of the previous budget year, the rate of increase climbed more rapidly as the user community gained confidence in the system's reliability. The dip at the end of calendar year 1968 resulted from the announcement of user charges as well as seasonal factors of the end of quarter and year-end holidays.

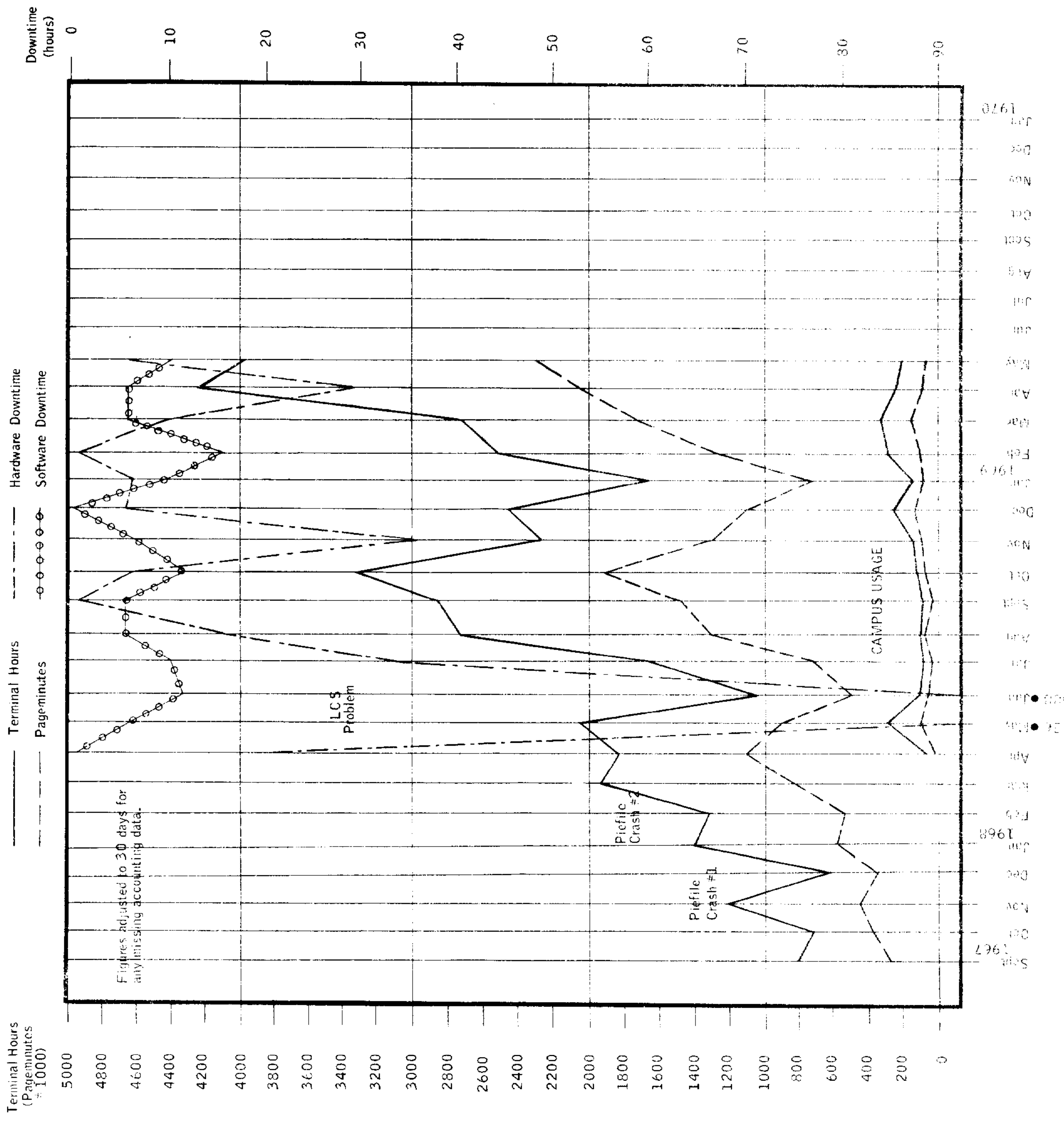
Grant No. FR00311-03
Section I-A

ACME Note AU-11

Gio Wiederhold/Charles Class
May 23, 1969

Monthly Usage at ACME

Absolute maxima of terminal hours per month: 10,170 (30 lines, 30 days, 11.5 hours per day). Absolute maxima of terminal page minutes per month: 7,866,000 (380 pages, 30 days, 11.5 hours per day).



Revision of AU-10 dated April 24, 1969.
Dist: Staff/All

Utilization as expressed in page minutes reached an all time high of 2,260,596 in the accounting period April 17, 1969 through May 16, 1969. Only 269,364 page minutes were used by the ACME staff and users accounted for almost two million. Approximately half of the two million was used by researchers with NIH grants which are presently exempt from users charges because they have non-competing renewal grants.

To maintain a cooperative relationship that is conducive to an effective operation, constant communication with the user is vital. The lightbox on the users' terminals provide both system and operations information to him. Daily messages at the time of log on inform him of changes in system and schedule; special broadcasts announce current system events that may affect users. An informed user is a happy user.

New features not yet documented in the manual are described in ACME User Notes.

ACME hours are from 7:00 a.m. to 5:00 p.m. and 6:30 p.m. to midnight, daily. General system development work and file maintenance, and four hours a week of IBM preventive maintenance, cover the remainder of the 24-hour-day operation. Exceptions may be made to use the period from 3:30 to 5:00 p.m. and 10:00 p.m. to midnight when urgent system work requires it.

When a shortage of terminal lines develops, the following priorities are observed:

First Priority: Users signed up for realtime experiments.

Second Priority: All other known users, queued in first-in, first-out order.

Third Priority: "Unknown" users (i.e., medical student terminals), only if there are free terminal lines.

The ACME TV display is scheduled by a separate sign-up sheet.

The operator has control over real-time usage to insure that the required response rates are not impacted by excessive usage. An effort is being made to train our own operators, including recruits from the local community.

A Sanders 720 graphic display unit (provided by the Genetics Department) allows the computer operator to obtain a list of users on the system at any given moment. It has increased our access to user status over the IBM typewriter terminal by a 15-to-1 ratio (from 150-second access to 10-second access).

Our new switchboard has semi-automated the connecting of user terminals to ACME. Formerly the user would telephone the computer room and request that a 2702 port be connected to his terminal. Our new board requires the user only to press the button on the lightbox. When the operator plugs an available port into the user's outlet, the indicator lights turn off.

Administrative Changes

In this third year of the ACME project Joshua Lederberg, Ph.D., Professor, and Chairman, Department of Genetics, continued as Principal Investigator. The ACME Policy Committee is drawn principally from the community of users in the Medical School. ACME is administered technically as a facility of the Stanford Computation Center.

During the year covered in this Report, Paul Armer succeeded Edward Feigenbaum as Director of the Stanford Computation Center. Professor Feigenbaum is concentrating on his teaching and research in the Computer Science Department, but continues to serve ACME as an advisor for computing techniques. Gio Wiederhold continued as Associate Director of the ACME Facility.

During the early months of the current year the trial period of the Stanford Computation Center Campus Facility attempting to market ACME service to non-Medical School users at Stanford was completed. Since this service offering did not attract enough utilization to offset the costs to the Campus Facility, their sharing of ACME costs was reduced substantially at the end of November, 1968. This support was reduced to the incremental cost of the bulk core for the balance of ACME's third year. This reduced support caused substantial rebudgeting of ACME funds as will be evident in the financial section of this Report.

User charging based on page minute utilization and file storage was implemented in April, 1969. The users were first notified of the impending charges in the Fall of 1968 and a first schedule of rates proposed to the Special Research Resource Board in November, but it took until April to work out the details of the charging mechanism. It is too early to assess the full impact of this policy change; but in general the Medical School Researchers are being quite cooperative in coping with the additional paper work necessary to obtain the administrative approvals and to request supplemental funding. As might be expected with the prospect of having to use their own research funds for computation, the users are devoting considerable attention to improving the efficiency of their program - especially in the use of disc storage. We feel this is important. The users are urged to consult with the ACME staff. With more efficient user programs ACME will be able to continue to serve promptly a growing number of users.

The following ACME schedule of rates for service was approved by SRRB by Dr. Raub's letter of April 8, 1969:

Effective Date	MARCH 21, 1969
Computer Services:	
Memory utilization of IBM 360/50 system:	
1. Research Service-Real Time, high interaction rate	1¢ per page minute
2. Research Service-routine terminals	2¢ per page minute
3. Administrative and patient care services, services to non-research users	3¢ per page minute
File space utilization of IBM 2314 direct Access Storage Facility	10¢ per block of disk storage per month
Consulting and Programming Services-ACME staff	no charge

ACME Education

ACME offers a beginning and an advanced course in PL/ACME programming. The beginning course describes the ACME system, the use of the terminal, and the PL/ACME statements for terminal input and output, calculation, and filing. The advanced course describes variation and options. Real-time (1800) use is not covered.

A beginning or advanced course requires three sessions of 1-1/2 hours. There are about ten students in a class. Of the 130 beginning students taught since July, 1968, 25 percent have been physicians, 30 percent students (medical, graduate, undergraduate, and some high school students), 25 percent lab technicians and secretaries, and 20 percent other researchers, administrators and others. Since the teaching program began in 1966, there have been 742 students (594 beginning, 148 advanced).

The current students generally plan to use ACME for statistical studies on large groups of patients and for processing laboratory data. Most realtime users took the course sometime earlier.

The response to the course is gratifying. Beginners discover that they can use a powerful machine. Many sign up for the advanced course and recommend the courses to their associates. Even students (particularly medical students) who do not program after the course feel that they have seen how a computer can help them later in their careers.

ACME PROGRAM LIBRARY--ON PUBLIC FILE

AH-1	ACME Program Library HELP: Information on ACME Keywords (G. Sanders) May 24, 1968
BLY-1	Clinic Patient Scheduling (Crouse) June 23, 1967
BQC-1	Questionnaire Program (Sanders) Apr. 15, 1968
DDA-1	Pediatrics Project--Routine No. 1 (Drew) Aug. 8, 1967
EAM-4	ACME Program Library LACKFIT: Test for Linearity of Regression (Schach) Feb. 7, 1969
EAN-4	ACME Program Library MULT: Multiple Regression (Moore/Schach) Feb. 17, 1969
EAP-3	ACME Program Library GENCORR: Correlation Coefficients--Missing Data (Kraemer) Feb. 7, 1969
EBD-3	ACME Program Library WEIGTREG: Weighted Linear Regression (Schach) Feb. 7, 1969
EBE-2	ACME Program Library LINREG: Linear Regression (Schach/Liere) Feb. 17, 1969
EBH-1	ACME Program Library ONCALL: Scheduling Program for Residents on Call (Moore) Jan. 8, 1968
EBI-3	ACME Program Library PCPLOT: Frequency Plot (Moore) Feb. 7, 1969
EBK-3	ACME Program Library POLY: Polynomial Regression (Moore) Feb. 7, 1969
EBL-2	ACME Program Library RUNGK_1: Runge-Kutta Solution of First-Order Ordinary Differential Equation (Liebes) Aug. 21, 1968
EBM-3	ACME Program Library ZEROFIT: Least-Squares Line through Origin (Schach) Feb. 27, 1969
EBN-2	ACME Program Library BSORT: Sorting (Liere) Sept. 18, 1968
EBO-2	ACME Program Library PEEL: Exponential Curve Fitting (Slimick/G. Sanders) Feb. 17, 1969
EBP-2	ACME Program Library KWTEST: Non-Parametric Analysis of Variance--One-Way (Kraemer) Jan. 12, 1969
* EBQ-4	ACME Program Library PLOT: Scatter Plotting (Liere) Mar. 27, 1969
EBR-3	ACME Program Library SCHUSTER: Schuster Periodogram (Schach) Feb. 7, 1969
EBS-2	ACME Program Library RUNGA6: Runge-Kutta Integration (G. Sanders) May 2, 1968
EBU-1	ACME Program Library TIMESER: Spectral Analysis (Schach) June 19, 1968
EBV-1	ACME Program Library GOODFIT: Test for Goodness of Fit (Schach) June 24, 1968
EBW-2	ACME Program Library DISCRIM2: Discriminant Analysis for Two Groups (Schach) Feb. 7, 1969
EBX-2	ACME Program Library TSQUARE: Hotelling's T Square (Schach) Feb. 7, 1969
EBY-2	ACME Program Library CHI_2by2: Chi-Square Statistic with Continuity Correction (Schach) Feb. 7, 1969
EDA-1	ACME Program Library MAPIT: Mapping Bacterial Chromosomes (Nye) Aug. 5, 1968
EDB-2	ACME Program Library DATAPROG: Writing a Data File Into a Program File (Liere) Sept. 17, 1968

	EDC-1	ACME Program Library FOURIER: Fourier Analysis (Liere) Aug. 5, 1968
	EDD-1	ACME Program Library HEXARITH: Hexadecimal Arithmetic Routines (Feinberg) Aug. 3, 1968
	EDE-2	ACME Program Library JUSTIFY: Text Justification (Emmons/Liere) Jan. 12, 1969
*	EDF-2	ACME Program Library RUNGK_2: Runge-Kutta Solution of Second-Order Ordinary Differential Equations (Liebes) Apr. 18, 1969
	EDG-1	ACME Program Library POWELL: Fitting Program for Nonlinear Functions (G. Sanders) Jan. 17, 1969
	EDH-1	ACME Program Library LISTER: Listing the User's Program (Liebes) Sept. 19, 1968
	EDI-2	ACME Program Library MATCH: Matching Donors to Recipients for Transplants (Bauriedel/Liere) Feb. 17, 1969
	EDJ-1	ACME Program Library LINSYS: Solution of Simultaneous Equations (Jones) Sept. 13, 1968
	EDK-2	ACME Program Library ANOVATWO: Two-Way Analysis of Variance--Unequal Cell Frequencies (Brast) Feb. 7, 1969
	EDL-1	ACME Program Library EDITER: Converting a Program to a Standard Format (Liebes) Nov. 14, 1968
	EDN-2	ACME Program Library BALTHREE: Analysis of Variance for a Balanced Three-Way Design (Kraemer) Feb. 7, 1969
*	EDP-2	ACME Program Library COPIER: Reproducing a Complete or Partial Program Data Set (Bassett) Jan. 8, 1969
	EFA-1	Translating FORTRAN Programs to PL/ACME Using DATAPROG, UNEKEVAR and TRANSLATE (Emmons) Sept. 23, 1968
	EFC-1	ACME Program Library UNEKEVAR: Unique Variables (Emmons) Sept. 30, 1968
	EFD-1	ACME Program Library TRANSLATE: Translation of FORTRAN Programs to PL/ACME (Emmons) Sept. 30, 1968
	TR-1	Respiratory Project on the 1800 (Hintz) Mar. 21, 1967
	TRA-1	Interfacing a Packard 3314 Scintillation Counter to ACME (Morris) Feb. 5, 1968. Summarized briefly in ACME Note BAE.
	WTX-1	Text Processing Routines (Wiederhold) Not dated.

ACME Note

ACME File Input/Output

FIO-1
Serge Girardi
May 8, 1969

The ACME file system provides in a time-sharing mode all the I/O facilities of the PL/I language with a few exceptions [1]. For ease of access and flexibility, the file is composed of system data sets and user data sets:

(1) System Data Sets

These data sets are accessed by direct block addressing rather than using the index facility. They include system information such as the catalog of user NAMEs and PROJECTs, the available space on each storage unit, and the directory indices pointing to a directory record for each cataloged data set.

(2) User Data Sets.

These data sets are composed of the collection of information--data or programs--manipulated by the users. Each data set is assigned a qualified name (User NAME. PROJECT. data set) by the cataloging facility and referenced through the use of an index and of a directory.

PHYSICAL FILE ORGANIZATION

System and user data sets are currently residing on eleven 2316 disk packs (Two IBM 2314 units). The present equipment would provide for 14 disk packs of data storage.

Disk Organization [2]

The IBM 2316 disk pack is composed of 11 disks and provides 20 surfaces on which data can be recorded. It is divided into 203 concentric cylinders numbered from 000 (outermost cylinder) to 202 (innermost cylinder), with 20 tracks per cylinder (00 to 19). Tracks in Cylinders 200 through 202 are alternate tracks that can be used if any tracks in Cylinders 000 through 199 should become defective.

Data Format

Each storage module is divided into fixed-length blocks (2000 bytes), three per track, except for tracks 0 through 5 of Cylinder 000 which are used by the Operating System. A module has a storage capacity of 3,994 tracks or 11,982 blocks, or 23,964,000 bytes.

Track Format

Figure 1 shows the track format.

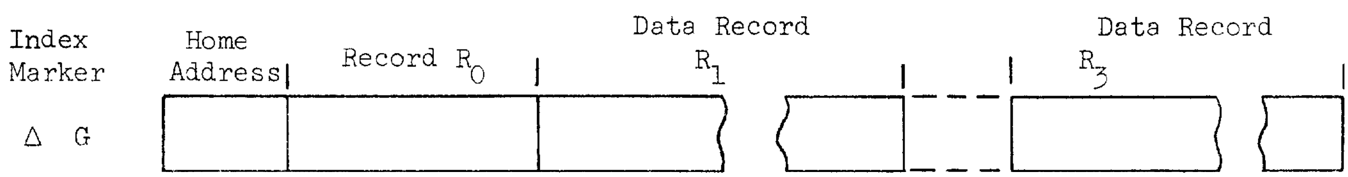


Figure 1. Track Format.

Track Format Subfields:

Index Marker: The beginning of a track is signalled when the index marker is detected. All tracks on a disk pack are synchronized by the same index marker (there is only one index marker per disk pack).

Gaps (G): Gaps separate areas on tracks and contain no data. Gap lengths vary depending upon location within a record and the record length. They permit switching from reading to writing between fields.

Home Address (7 bytes): The home address defines the condition and location of the track (Figure 2).

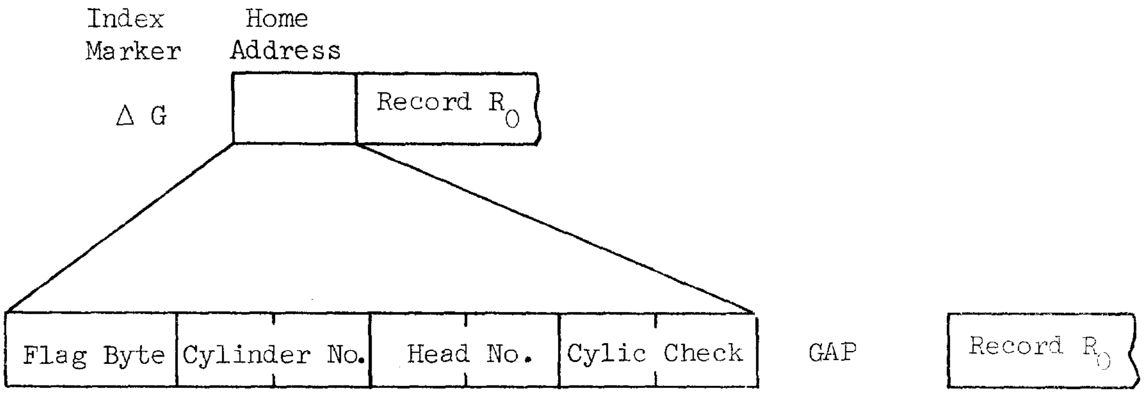


Figure 2. Home Address Format.

Track Descriptor Record (R_0): This is the first record following the home address, on each track. It is used by IBM programming systems to move the entire content of a track to alternate tracks if a portion of the primary track becomes defective. R_0 is divided into the same fields as data records except that it is not preceded by an address marker. It has a length of 21 bytes.

ACME data Records (R_1-R_3): These three data blocks are used by ACME data sets; all the other fields are initialized and used by IBM programming systems. Figure 3 gives the record format.

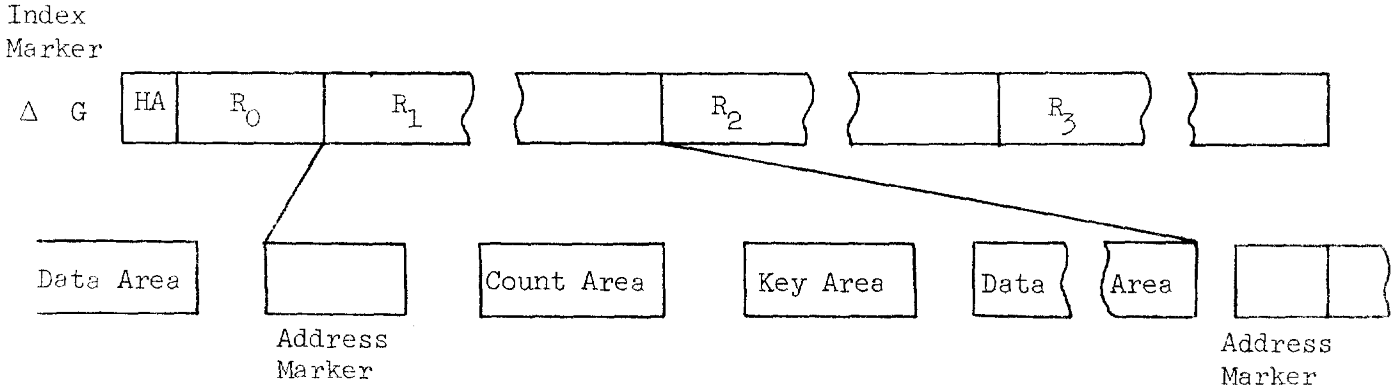


Figure 3. ACME Data Record Format.

Address Marker: This area indicates the beginning of each record. It is supplied by the 2314 control unit when the pack is initialized and contains a bit configuration that can be detected by the 2314 as the area preceding a count field.

Count Area (11 bytes): This field is divided into seven subfields (Figure 4).

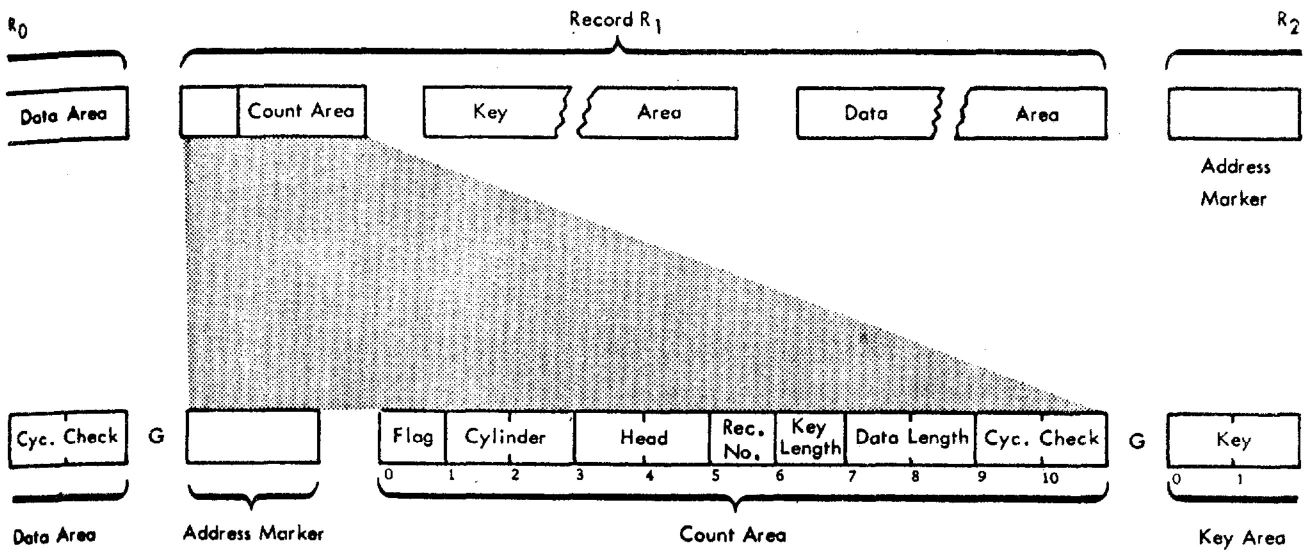


Figure 4. Count Area Format.

Count Area Subfields:

Flag (1 byte): Generated by the 2314 control unit as each record is initialized. If has information on the record number (odd or even), on the track condition, and on overflow records.

Cylinder (2 bytes): Byte 1 is set to zero. Byte 2 has the cylinder number from zero to 202.

Head (2 bytes): Byte 3 is set to zero. Byte 4 has the READ/WRITE head number (0 to 19) for the disk surface on which the record is stored.

Record Number (1 byte): Sequential number of the record on the track (1 to 3).

Key Length (1 byte): Number of bytes excluding cyclic check bytes in the key area: 8

Data Length (2 bytes): Number of bytes excluding cyclic check bytes in the data area: 2000.

Cyclic check (2 bytes): Used for cyclic checking information in the count area.

Key Area (10 bytes): This field contains part of the count field information (cylinder, head, record number), the data set number of the data set to which the block is assigned, and cyclic check information. (Figure 5). This field is essentially used for block identification in addition to the standard information provided by the count field.

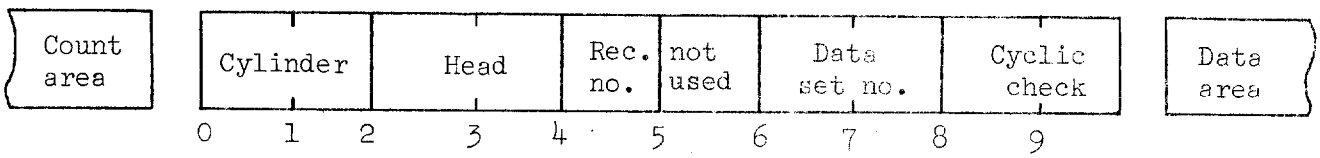


Figure 5. Key Area Format.

Data Area (2002 bytes): The data block is the smallest unit of space that can be assigned to a data set. It has a fixed length of 2000 bytes plus 2 bytes of cyclic information provided by IBM programming systems (Figure 6). Variable-length header and trailer contain several flags for differentiation purposes (system or user blocks), a data set number, a block address, and record length [3], [4].

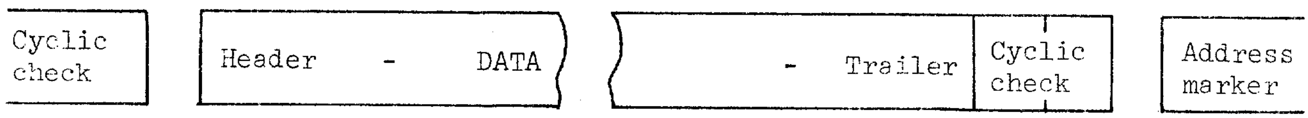


Figure 6. Data Area Format.

ACCESS METHOD

The ACME file I/O system uses the execute channel program (EXCP) macro-instruction to read or write data blocks [5]. This low-level access method provides the system with greater control over the I/O operations than the standard access methods. The EXCP macro-instruction uses the Operating System functions that provide for scheduling and queuing I/O requests, interruption procedures, error recognition, and retry. It passes control information to the I/O supervisor regarding a channel program to be executed. The file I/O system is only concerned with setting up a sequence of channel commands for the desired I/O operation

Block Addressing

Reading or writing a data block requires a positioning of the access mechanism (2314 unit) of the selected pack to the proper cylinder, track, and record on the track. Each block is assigned a number from 0 to 11,981 (maximum pack capacity), and in order to locate a block it is only necessary to specify a 32-bit address. (Figure 7).

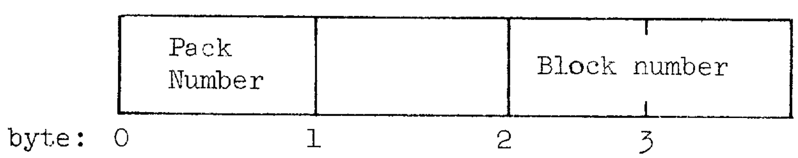


Figure 7. Block Address.

From this information a simple conversion gives the hardware address in terms of pack, cylinder, head, and record number.

Space Management

When a new storage unit (2316 disk pack) is being added to the file system, an initializing program formats all the blocks in the pack with the proper count and key field information and writes zeros in the data area. Block addresses in the form of block numbers are then stored in the space data sets. When creating new data sets or expanding existing ones, a routine picks up available blocks from the space data sets and assigns them. When data sets are deleted, the freed blocks are returned to the space list either on-line or by a stand-alone file analyzer.

A block with the available status has a data set number in its key field equal to the space data set number. An efficient data protection is achieved by testing this information in the channel program prior to writing records in the data field.

Input/Output Operations

Three types of I/O operations are required in the ACME file system. Each of them leads to the coding of a different channel program. The operating system handles the positioning of the access mechanism to the proper cylinder on a pack and the selection of the desired head. Upon a successful completion of this SEEK command, control is passed to the supplied channel program. Its first command is a search identifier (SEARCH ID): a comparison is made between 5 bytes of data from CPU storage (cylinder, head, and record number) and the 5-byte record identifier portion of a count area from the storage unit. Since a track has three data blocks, it may be necessary to reissue this command until a match occurs (Transfer In Channel command).

Writing a Block. A WRITE operation is always preceded by a test on the key area (SEARCH KEY EQUAL command). In this case the block must be available and the data set number in its key area reflects this status. If no match occurs an error is signalled and the block is not written. With a successful match, additional commands reposition the writing head by disk rotation after the count field and write the key and data area from the user buffer. The data set number in the key field is written as the data set number of the data set to which the block is assigned. Appendix A gives an example of the sequence of commands.

Reading a Block. After positioning of the access mechanism, the key and data fields are read into a buffer in the user region. No testing is made on the key field since a data protection on a READ operation is not necessary. A check for valid information in the buffer is made before data transmission to the user.

Rewriting a Block. This operation is similar to writing a block except that they key field of the data record is not modified. Data protection is achieved by comparing the key field with the key in the buffer from which the block is rewritten.

Appendages

These routines are entered by the I/O Supervisor upon successful completion of a channel program or detection of an error.

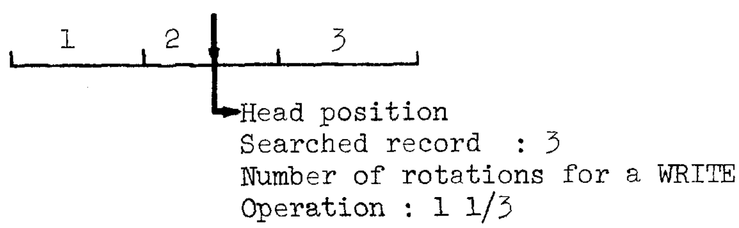
Normal End Appendage. This routine signals a normal completion of an I/O operation to the ACME system posting table. Control can then be given to the user.

Abnormal End Appendage. This routine is entered twice whenever an error is detected during execution of a channel program. The first time entered, the file system checks for two types of errors: NO MATCHING KEY which can occur on a WRITE or REWRITE operation, and NO MATCHING ID which can occur during any I/O operation. With other types of errors (DATA CHECK, OVERRUN, etc.) the IBM-supplied error recovery routines attempt to restart the channel program several times. If the error is permanent, the appendage is entered a second time and the ACME system posting table is posted with the permanent error condition, which is signalled to the user when he receives control. Errors like NO MATCHING KEY can be caused by the ACME system for instance when a block supposedly available is in fact used by a data set. No automatic error recovery is attempted in this case because of the overhead involved; instead system programmers can patch the damaged information with a file fixer program while the ACME system is in operation.

PERFORMANCE

Most of the time required to service an I/O request is spent in mechanical motions. Access time varies from 135 ms for a SEEK between the extreme inner and outer cylinders to 25 ms for a SEEK between adjacent cylinders. Figure 8 shows the minimum and maximum time spent in mechanical motion for a WRITE operation, with 25 ms per disk revolution [2].

The minimum time occurs when the selected read/write head is positioned just before the address marker of the searched record (Figure 9a).



Firgure 9a: Head Position When Minimum Time is Spent for a WRITE Operation.

The maximum time is spent when the head has just passed the address marker:

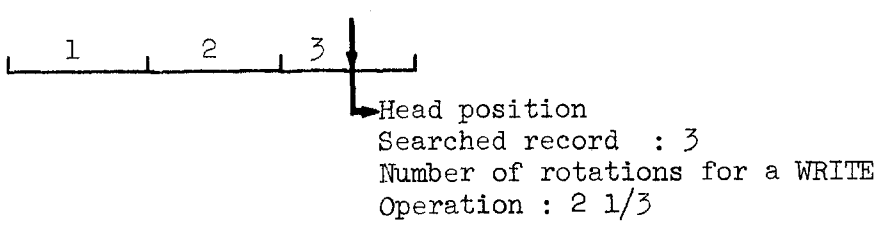


Figure 9b: Head Position When Maximum Time is Spent for a WRITE Operation.

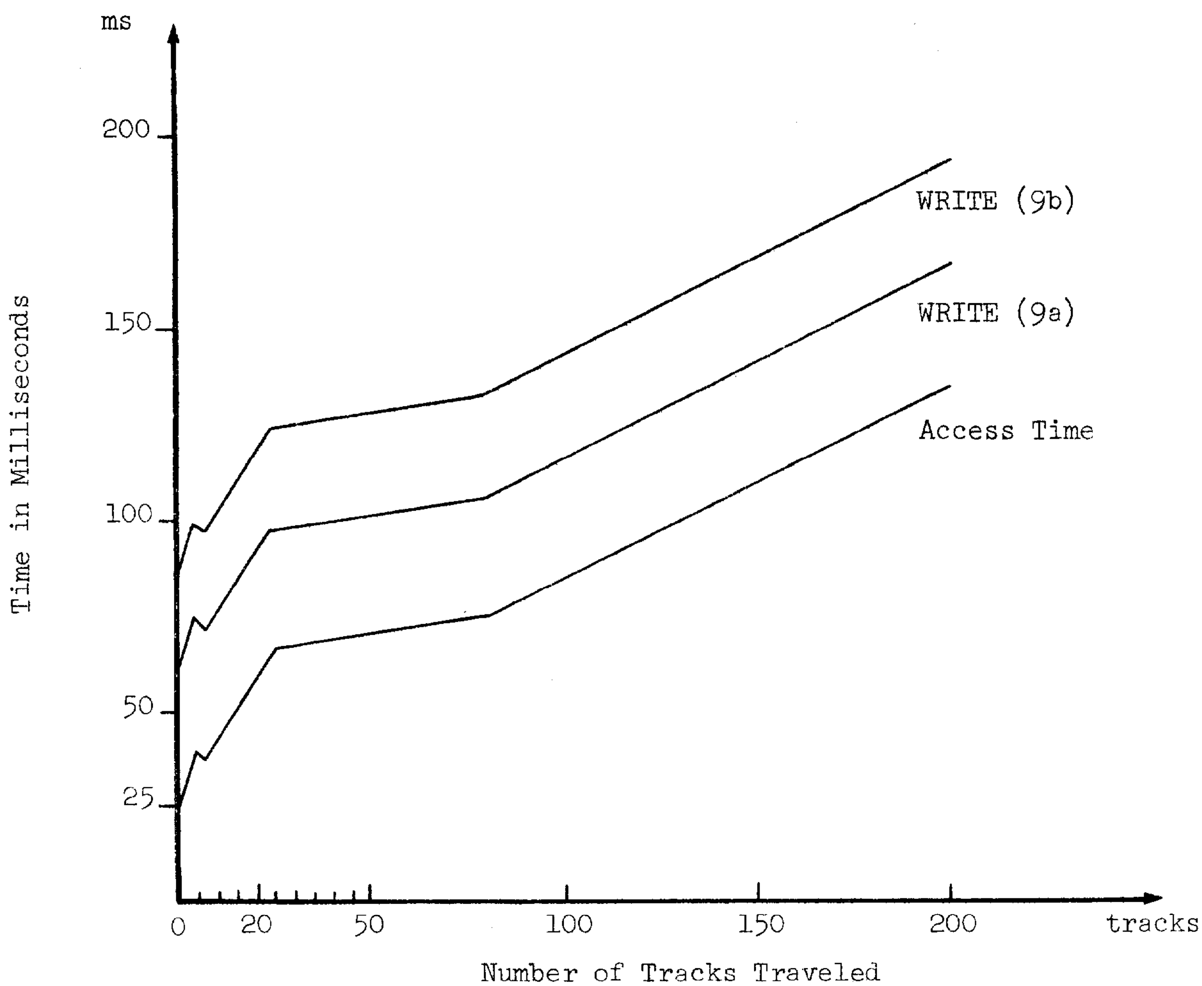


Figure 8. Mechanical Motion Time for a WRITE Operation.

APPENDIX A

Channel Program for a WRITE Operation

```
    SEARCH ID
    TIC * -8
    SEARCH KEY EQUAL
(a) TIC * -16
(b) READ DATA
    SEARCH ID
    TIC * -8
    WRITE KEY and DATA
```

(a) This Transfer In Channel command to another TIC will generate a program check in case no match occurs in comparing the KEY field immediately following the COUNT field of the searched record with the supplied KEY data. The error condition is detected in the Abnornal End Appendage.

(b) This is a dummy READ required by the control unit operations when SEARCHing on the same track more than once.

FOOTNOTES

1. Miller, Jerry, "The ACME File System," ACME Note FY-1, February 27, 1969.

2. IBM System 360 Component Descriptions--2314 Direct-Access Storage Facility and 2844 Auxiliary Storage Control, A26-3599.

3. Miller, Jerry, "ACME File System--Data Sets," ACME Note FA-5, September 27, 1968.

4. Frey, Regina, "ACME File System--Codes," ACME Note FC-2, August 5, 1968.

5. IBM System/360 Operating System. System Programmer's Guide, C28-6550.

6. Girardi, Serge, and Jerry Miller, "ACME File System--Control Block Formats," ACME Note FB-3, March 31, 1969.

Dist: Prog/All

SUMMARY OF RESOURCE USAGE

Month and Days	Daily Scheduled Service	Account Records Console Hours	Account Records Pageminutes
Apr 21 to May 20	700-1530 1830-2200	2,075	947,000
May 21 to June 20	700-1530 1830-2200	1,003	467,009
June 21 to July 20	700-1530 1830-2200	1,626	712,514
July 21 to Aug 20	700-1530 1830-2200	2,761	1,313,940
Aug 21 to Sept 20	700-1530 1830-2200	2,856	1,481,671
Sept 21 to Oct 20	700-1700 1830-2400	3,330	1,955,295
Oct 21 to Nov 20	700-1700 1830-2400	2,262	1,281,133
Nov 21 to Dec 20	700-1700 1830-2400	2,461	1,190,110
Dec 21 to Jan 20	700-1700 1830-2400	1,639	738,048
Jan 21 to Feb 20	700-1700 1830-2400	2,521	1,213,289
Feb 21 to Mar 20	700-1700 1830-2400	2,757	1,714,973
Mar 21 to Apr 20	700-1700 1830-2400	4,108	2,010,110

USE BY ACME STAFF (CORE PROJECTS)

project	department	account	runs	hours	pageminutes(K)	blockmonths(K)	months (1)	months (2)
LHALLEN_.T124LAHN	(*SCC	) T124-7	1	:14	0.068	0.060	7	6
PLBASSET.ACMECONS	(*ACME	) L001-7	353	395:12	168.389	3.43200	7	6
BEEBE,RJ RJBEEBE_.T177TRLL	(*SCC	) T177-7	43	19:53	7.92300	.151000	12	6
BERNS,RI RIBERNS_.T178BERN	(*SCC	) T178-7	100	40: 0	21.2400	.200000	12	6
BREITBARD,G G_BREITB.ACME	(*ACME	) L002-7	330	147:51	42.3350	.339000	12	6
CLASS,C C_CLASS_.ACME	(*ACME	) L004-7	1093	1959: 9	508.040	.634000	12	6
CROUSE,L L_CROUSE.CATH_LAB	(*ACME	) L005-7	110	127:48	55.8120	8.62900	12	6
CUMMINS,D D_CUMMIN.DOMESTIC	(*ACME	) L006-7	85	42:42	10.5130	.187000	12	6
DELAROCA,D D_DELARO.IBM1800	(*ACME	) L007-7	110	108:45	88.6000	.848000	10	6
EMMONS,K K_EMMONS.ACME	(*ACME STAFF	) ....-7	246	256:44	119.463	0.000	7	0
FEINBERG,DA DAFEINBE.ACME	(*ACME	) ....-7	91	17:50	6.23900	0.000	7	0
FREY,R r_frey__.acme	(*ACME STAFF	) L008-7	192	102:46	43.8210	4.09510	6	6
FREY,R r_frey__.null	(*ACME	) L008-7	188	89:33	25.7030	0.000	12	0
GILMAN,J J_GILMAN.ACME	(*ACME	) ....-7	3	1: 1	.209000	0.000	7	0

(1) number of months using system

(2) number of months of file accounting data